

Multipurpose mobile platform for telemedicine applications

Timo Laakko, Juha Leppänen, Jaakko Lähteenmäki, and Antti Nummiaho
VTT Technical Research Centre of Finland, Espoo, Finland

Abstract—A new mobile platform is described that can be used in a number of telemedicine and wellness applications to provide connectivity between healthcare professionals, patients and measurement devices. The platform is based on an open application framework that provides interfaces for measurement, user interface, database and network connectivity implementations. A new technical approach has been adopted in using a mobile database solution for local storage and caching of information at the mobile terminal. Further, the platform applies a document-based service architecture that provides a versatile and reliable way of sharing and collaboratively complementing of health information including standard information model documents (HL7/CDA), and integration with health information systems and personal storages. The implemented mobile tele-ECG case demonstrates the overall function of the platform.

Index Terms—eHealth, mobile healthcare, mobile platform, telemedicine applications

I. INTRODUCTION

Telemedicine is increasingly gaining popularity due to its high potential for cost savings and increased efficiency in healthcare. Although, there are only a few examples of operational services, rapid growth in the area is expected [1]. Mobile healthcare applications [2], [3] are emerging. The systems typically include devices for medical measurements and communication platforms for information sharing. Interfaces are in most cases proprietary and, consequently, the health service provider is bound with one particular device manufacturer. The situation is expected to change in the future. Large companies are pushing interoperable solutions through

industry consortia, such as the Continua Health Alliance [4], and open interfaces [5] enable multipurpose platforms.

This paper describes an open mobile platform, UPHIAC (Ubiquitous Personal Health Information Access), which can be used in a number of applications to provide connectivity between healthcare professionals, patients and measurement devices (Fig. 1).

II. THE UPHIAC MOBILE APPLICATION FRAMEWORK

The UPHIAC application framework includes open interfaces for measurement, storage, network connectivity and user interface implementations (see Fig. 2). Further, the UPHIAC platform provides implementation of core classes that are common for all implementations of the interfaces and constitute basic functionality of the platform. A new UPHIAC application is created by providing implementations for the core interfaces. The platform includes default implementations of the interfaces, so new implementations of all the interfaces are not required. In principle, the platform works as an application manager, which reads data from one interface and writes it to another interface. The interfaces connect the platform to the outer world.

UPHIAC documents are the main data entities handled by the platform. In that, the document-based approach of the PIR (Personal Information Repository) architecture [6], [7] has been followed. A document is considered as a measurement data augmented with additional metadata such as information about the patient (e.g., identity number). The raw data from measurement device is completed into the UPHIAC document form. Further, the document provides the means for sharing, analyzing and collaboratively completing health information.

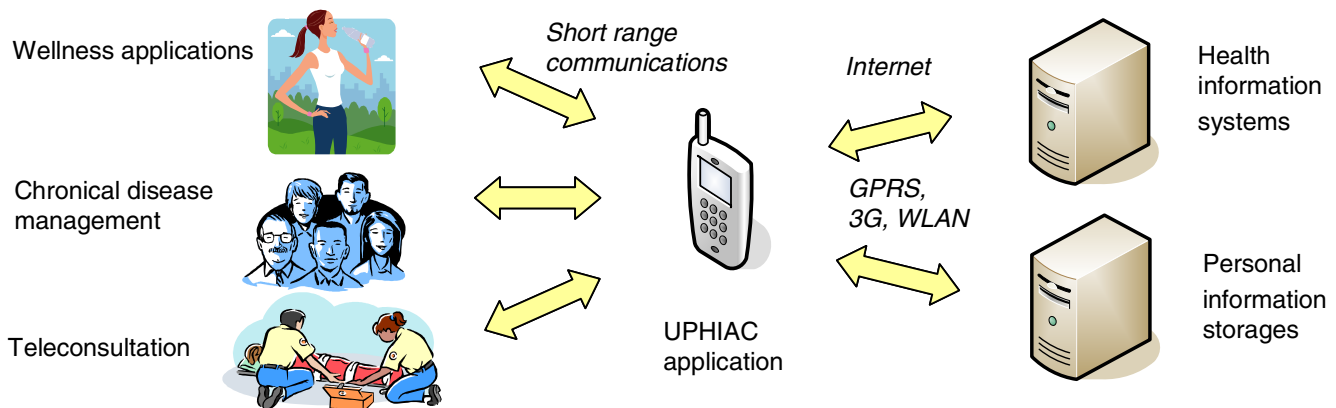


Figure 1. Mobile platform providing connectivity between distributed measurement devices and information systems.

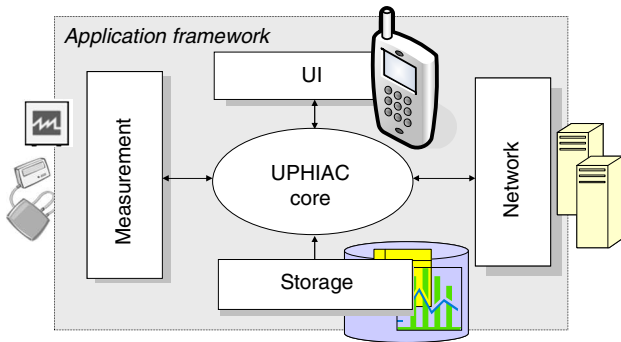


Figure 2. The UPHIAC application framework.

The core interfaces (cf. the sub interfaces of the abstract base interface UPHIACInterface in Fig. 3a) include:

- **Measurement.** This interface communicates with measurement devices. Each device provides a measurement service that constitutes an entry point for data transfer and measurement.
- **Network.** The data are shared and synchronized with a virtual storage space located in the server side (external health information system or personal storage). It is also possible to send and receive alerts within the virtual space.
- **UI.** The interface takes care of user interaction and visualizing defined data, e.g., as tables or graphs. It can be adapted suitable for different roles of authenticated users.
- **Storage.** The storage interface takes care of the local storing of the measurement data and other information handled by the platform.

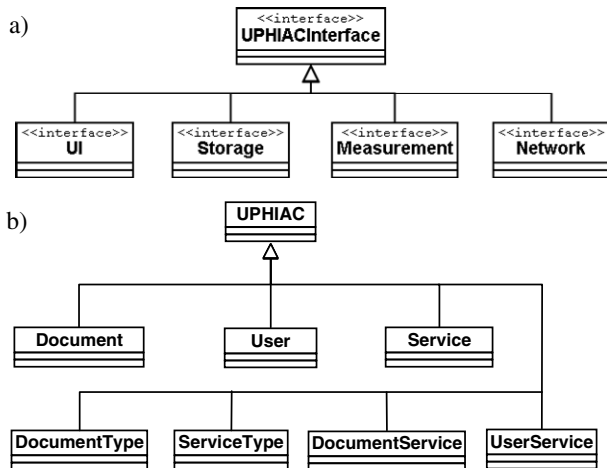


Figure 3. The core interfaces (a) and classes (b) of UPHIAC.

The core classes (cf. the subclasses of the abstract base class UPHIAC in Fig. 3b) are summarized in the following:

- The **Document** class represents an UPHIAC document.
- The **Service** class represents a service provided by health information systems and personal storages, and it can be applied to the related document in order to share and complete document information.
- The **User** class represents user and user role specific

information.

- **ServiceType** provides additional information related to a service, including information about which document types can be applied by the service.
- **UserService** specifies how a user can access and use a service.
- **DocumentType** provides additional information about the document types.
- **DocumentService** provides information about services applied to a document.

III. RETRIEVING MEASUREMENT DATA

A measurement service provided by a measurement device is used to start and stop measuring, and to retrieve and push measurement data.

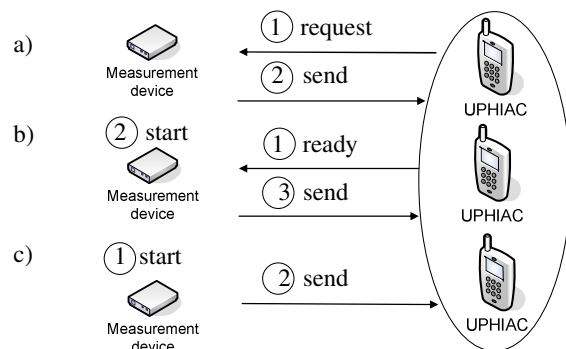


Figure 4. Retrieving data from measurement device to UPHIAC.

The retrieving of data can be accomplished in several ways (cf. Fig. 4). The control of starting the data transfer can be in either side, and the measurement can be initiated either by the UPHIAC application or by the measurement device. In the case a), first, the UPHIAC application makes a request for measuring. In the case b), the UPHIAC application indicates that it is ready for receiving measurement data. In the case c), the control is in the measurement device side.

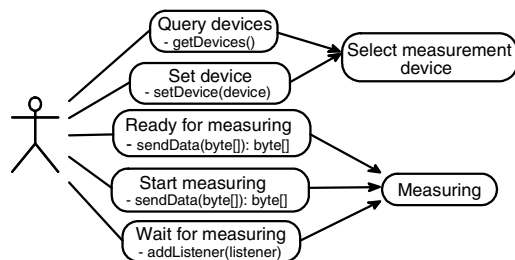


Figure 5. Measurement interface methods.

Reference implementations have been done for each of the above cases. In a demonstration configuration, the measurement application and the UPHIAC application were installed in separate mobile phones, and communication between the applications was accomplished using Bluetooth. In the implementation, the Java Bluetooth API (JSR 82) was used. Also, the Java MIDP push registry facility has been

applied to enable applications to be launched automatically, without user initiation. On one hand, the push registry is used to wake up the measurement application while the UPHIAC application is requesting or waiting for measurement, and, on the other hand, to start the UPHIAC application while the measurement application is directly pushing data.

The Measurement interface implementations are specific to the measuring devices, and the implementations (cf. Fig. 5) should provide methods for:

- sending and receiving raw data (bytes) to/from measurement device,
- querying measurement devices and setting device to be used, and
- getting name and address of the measurement service provided by the measurement device.

The querying measurement devices and selecting the device is required only when the measurement device is changed (presetting phase). The selected device is set as the current measurement device and stored for later use.

IV. LOCAL DATA MANAGEMENT

The UPHIAC storage platform implements the Storage interface of the framework and manages locally the measurement data enriched with additional information (metadata etc.) contained in the input data model. The information includes the services provided by external health information systems (and personal storages) and measurement services. Also, user-specific information such as user profile, role, and access rights to services are stored in the database.

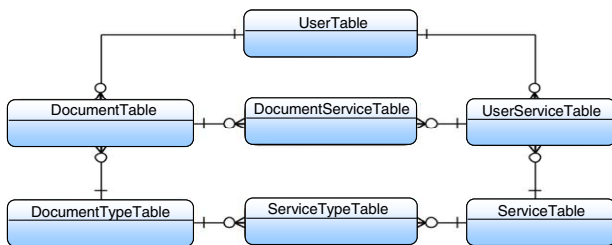


Figure 6. Outline of the UPHIAC mobile database schema.

Typically, the storage and processing capabilities of a mobile device are limited compared to that of desktop systems. Therefore, it should also be possible to transfer the data to external storages if required. Further, the mobile storage system needs to be as light as possible and, thus, not necessarily has to provide large general database functionality (such as full SQL database functionality), but more likely the operations that are required by the application suite. Also, the portability demands in providing implementations to a variety of different mobile devices should be noticed.

The UPHIAC database is defined so that it provides general local storage for the information handled by the diverse UPHIAC applications. Fig. 6 outlines the UPHIAC database schema, which can be implemented as a relational database (e.g., SQL database). The data fields of the tables have a close relationship to the contents (attributes) of the respective

UPHIAC core classes.

The storage interface (Fig. 7) provides an application interface to the local database, and all the database operations by UPHIAC applications are performed through it. There are 34 different interface methods. For instance, basic methods to store instances of each of the UPHIAC classes (`addDocument`, `addUser`, ...) and to retrieve instances from database (`getDocument`, ...). The interface aims to make the management of UPHIAC entities very flexible.

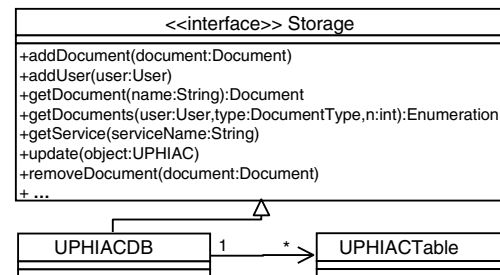


Figure 7. Storage interface.

The UPHIAC reference implementation of the Storage interface provides a light implementation of the UPHIAC database schema with relational database functionality for UPHIAC applications. It is implemented using the Java MIDP RMS (Record Management System). The database is shared within the UPHIAC application suite (may include several Java MIDLets that implement UPHIAC UI), but is not shared with other applications. In the database implementation, the payload data is stored with IDs and other metadata for finding and managing the data. A document service specifies the services applied to the document and may include references to external storages. Therefore, the storage platform is also aware of the real placement of each data element, even if located in an external storage.

V. INTEGRATION WITH EXTERNAL SYSTEMS AND STORAGES

In order to share the documents containing health information each party handling the documents must have a common understanding of the documents, their contents and structure and the usage roles. The integration can be enabled by the specification work done by HL7 [8] offering methods for exchanging, storing and accessing the patients' health information and parsing the contents into structured format.

The data can be stored into the hospital systems like Electronic Health Records (EHR) (cf. HL7 EHR [8]). When the patient's personal information (name, social security identity etc.) is added, the newly measured data can be linked with the patient's medical history, which is in key role in the patient's health care process.

UPHIAC can exchange documents also in a HL7 format that constitutes the base for the integration with external systems. A document can be completed step-by-step (cf. Fig. 8):

Step 1. Empty XML-structures are available at the server.

Step 2. When the user logs in, (s)he has a certain role and preferences. The user specific data is sent to the server, which

can then start filling in the user's data (*Provider Info*) that can also be used for providing the digital signature for the data.

Step 3. Each measurement instance produces telemedicine data, which is sent to the server. The measured data is used for updating the structured document (*Payload Data*).

Step 4. The document needs to be updated with the measurement target's overhead data (*Patient Info*).

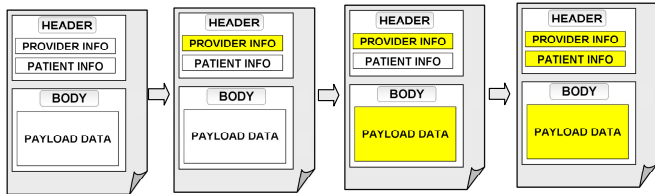


Figure 8. Incrementally complementing a telemedicine document.

In the typical usage case, the heart patient (who needs instant treatment) is taken to the hospital by an ambulance. In another case, the home care patient's well-being is being monitored. Both cases require wireless and automated connection [9] between the health sensors near the patient and the remote expert. Data sharing with the expert supports the collaboration for the benefit of the patient's treatment process.

VI. CASE EXAMPLE: TELE-ECG MEASURING

Tele-ECG refers to a use case, where the Electrocardiogram (ECG) of a heart patient is sent from the ECG measurement device to another location for the analysis carried out by a cardiologist. The use case involves the following steps:

- In the less critical phase, the UPHIAC platform components will be initiated in the moving treatment unit
- The actual measurement can be started in the moving unit from the measurement device. The measured data will be stored locally and delivered automatically to the remote server shaped into the structured form, namely into FDA XML based on HL7 version 3 messaging standard [10].
- The cardiologist receives the structured data and, then, analyses the graphs (Fig. 9), prepares the first conclusions, and send the results back to the moving unit.
- After or during the treatment phase in the moving unit the measured data will be complemented with the patient's personal information such as the social identity number.
- The cardiologist gets the complemented data and will be able to search more information about the patient from

EHR and prepare the patient's reception at the hospital.

The implemented Java MIDP example application (see Fig. 9) realizes the above steps. PIR document server [7] acts as the remote application server. Due to the proprietary solutions of the ECG meter interfaces it is currently not possible to use a commercial ECG measurement device. Instead another mobile phone emulates the meter functionalities for demonstrating the automatic delivery of the measured data from the meter to the remotely resided expert. It is however expected that in the near future the device interfaces will be opened for wider use allowing the real end-to-end integration.

VII. CONCLUSIONS

The design targets of the UPHIAC platform have been modularity, generality, reusability and extensibility. The open interfaces of the UPHIAC mobile platform facilitate implementations of different kinds of applications and connectivity with a variety of measurement devices and health information systems. The local database interface provides versatile and efficient methods for managing the core data entities. Further, the document-based service architecture approach enables the sharing and collaboratively complementing of health information and integrating the platform within organizational and personal storages.

REFERENCES

- [1] J. Edwards, "The potential of Telemedicine Applications", Gartner, Industry Research, ID G00142598, 16.10. 2006.
- [2] R. Istepanian, E. Jovanov, and Y.T. Zhang, "Introduction to the special section on M-Health: beyond seamless mobility and global wireless health-care connectivity", *IEEE Transactions on Information Technology in Biomedicine*. Vol. 8, no. 4, Dec 2004.
- [3] E. Kafena, K. Dickson, and S.C. Cheung, "Alerts in Mobile Healthcare Applications: Requirement and Pilot Study", *IEEE Transactions on Information Technology in Biomedicine*. Vol. 8, No 2, June 2004.
- [4] Continua Health Alliance. <http://www.continuaalliance.org/>
- [5] T. Broens, A. van Halteren, M. van Sinderen, and K. Wac, "Towards an application framework for context-aware m-health applications", in *Proc. of 11th Open European Summer School (EUNICE 2005)*, 6-8 July 2005, Colmenarejo, Spain.
- [6] PIR - Personal Information Repository. <http://www.pir.fi/>
- [7] J. Lähteenmäki, J. Leppänen, and H. Kaijanranta, "Document-based service architecture for communication between health and wellness service providers and customers," (submitted to Pervasive Health 2008).
- [8] Health Level Seven. <http://www.hl7.org/>
- [9] E. Strömmer, J. Kaartinen, J. Pärkkä, A. Ylisaukko-oja, and I. Korhonen, "Application of nearfield communication for health monitoring in daily life," in *Proc. of the 28th IEEE EMBS Annual Int. Conference*, New York, Aug 30 - Sept 3, 2006.
- [10] The OpenECG project. <http://www.openecg.net/>

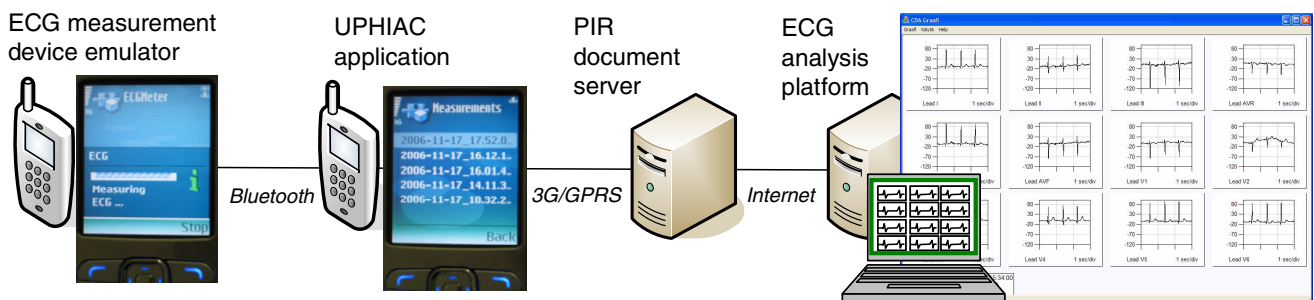


Figure 9. Tele-ECG measuring case.